Software Requirements Specification (SRS)

IEEE 830-1998 Format

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the system **"AI for Legal Document Summarization + Bias Detection"**. This system will automatically summarize legal documents such as contracts and court judgments and detect potential biases, such as gender disparities in rulings. It is intended for legal researchers, law firms, regulatory bodies, and ethics scholars.

1.2 Scope

This system will use Natural Language Processing (NLP) models, including BERT and transformer-based summarization algorithms, to analyze and produce concise summaries of legal texts. Additionally, it will detect linguistic and statistical biases in legal documents, highlighting potential ethical concerns such as gender bias in outcomes.

The key features include:

- Upload or paste legal text.
- Auto-summarization of legal documents.
- Bias detection (e.g., gender-based disparities).
- Highlighted outputs and downloadable reports.
- REST API for integration.

1.3 Definitions, Acronyms, and Abbreviations

- **AI** – Artificial Intelligence
- **BERT** – Bidirectional Encoder Representations from Transformers
- **NLP** – Natural Language Processing
- **API** – Application Programming Interface
- **SRS** – Software Requirements Specification
- **Bias Detection** – Identification of unfair patterns, e.g., gender or racial bias, in data or text

1.4 References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- Devlin et al., 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*
- Legal NLP Datasets: CaseHOLD, ECtHR, etc.

1.5 Overview

This document is organized to describe the system's functional and non-functional requirements, external interface needs, system features, and other relevant constraints or assumptions.

2. Overall Description

2.1 Product Perspective

This is a standalone system but can be integrated via API with existing legal management tools or research platforms. It uses pre-trained transformer models and custom regex-based rules for bias identification.

2.2 Product Functions

- Ingest legal text (PDF, DOCX, or raw text).
- Generate a concise summary using transformer models.
- Detect and highlight potential biases.
- Provide visualization and downloadable reports.
- Support RESTful API calls.

2.3 User Classes and Characteristics

- **Legal Researchers**: High familiarity with legal jargon, need detailed summaries.
- **Law Firms**: Require fast, accurate summarizations and bias alerts.
- **Ethics Scholars**: Focused on fairness, bias metrics, and documentation.
- **Regulators/NGOs**: Need interpretable results and audit trails.

2.4 Operating Environment

- Web interface: ReactJS + Flask backend
- Model servers: Python 3.x, PyTorch, HuggingFace Transformers

- Supported formats: .pdf, .docx, .txt
- Cloud-compatible: AWS/GCP/Azure

2.5 Design and Implementation Constraints

- Use of pre-trained NLP models (e.g., BERT, T5, GPT-based summarizers).
- Must adhere to privacy laws (e.g., GDPR) regarding document storage.
- Summarization output should retain critical legal meaning.
- Bias metrics must be interpretable.

2.6 User Documentation

- User Manual
- API Documentation
- Sample Reports and Bias Detection Guide

2.7 Assumptions and Dependencies

- Availability of clean legal datasets for fine-tuning.
- Users provide structured or well-formatted documents.
- External libraries (Transformers, spaCy) are maintained and updated.

3. Specific Requirements

3.1 Functional Requirements

FR1. Document Upload

- The user shall be able to upload documents in PDF, DOCX, or TXT format.

FR2. Summarization

- The system shall generate a summary containing ~15–25% of the original content length using a transformer model.

FR3. Bias Detection

- The system shall analyze documents and flag potential biases based on gender terms, case outcomes, and frequency statistics.

FR4. Visualization

- The system shall display summaries and highlight bias-related terms or patterns in the text.

FR5. Reporting

- The user shall be able to download a report containing the summary and bias analysis in PDF format.

FR6. API Access

- The system shall expose endpoints for summarization and bias analysis for integration.

3.2 Performance Requirements

- The system shall process and summarize a 10-page legal document in under 30 seconds.
- Bias detection shall complete in under 20 seconds for similar documents.

3.3 Logical Database Requirements

- Metadata: Filename, upload timestamp, user ID
- Summary Data: Text output, compression ratio
- Bias Logs: Detected entities, terms, bias score

3.4 External Interface Requirements

3.4.1 User Interface

- Clean, minimal web UI with upload panel, result viewer, and export options.
- Highlighted summary with tooltip explanations for bias flags.

3.4.2 Hardware Interfaces

- Web-based; no direct hardware dependencies.

3.4.3 Software Interfaces

- Python backend with Flask or FastAPI
- HuggingFace Transformers
- SQLite/PostgreSQL for metadata

3.4.4 Communications Interfaces

- HTTPS REST API

- JSON input/output format

4. Other Non-functional Requirements

4.1 Security

- All uploads must be encrypted (SSL).
- User data should not be stored unless explicitly allowed.

4.2 Reliability

- Uptime > 99.5% on cloud deployment
- Fallback mechanisms for failed summarizations

4.3 Maintainability

- Modular architecture to update models and bias detection rules easily.

4.4 Portability

- Compatible across all modern browsers.
- Containerized for deployment on cloud/VPS environments.

5. Appendices

A. Bias Detection Strategy (Simplified)

- Named Entity Recognition (NER) for gendered names/pronouns
- Regex rules for biased language
- Statistical outcome analysis (e.g., frequency of verdicts by gender)

B. Example Datasets

- ECtHR dataset
- Indian Supreme Court case archives
- ContractNLI for clauses